(c) writing at least one of moving object definitions, background character definitions and bitmapped graphics to a 96 kilobyte video random access memory storage; and

(d) generating a game display that is responsive to said user-manipulable control and is based at least in part on the contents of the object attribute memory storage, the color palette random access memory storage, and the video random access memory storage.

38. The method of claim 37, further comprising storing bitmapped color information for the simultaneous display of up to 32,768 different colors into said video random access memory storage.

39. The method of claim 37, further including referencing the color palette random access memory as a color lookup table for bitmapped graphics stored in said video random access memory storage to provide simultaneous display of up to 256 different colors.

40. The method of claim 37, further including defining the video random access memory storage to begin at address 06000000h.

41. The method of claim 37, further including allocating two frame buffers in the video random access memory storage to provide full motion video.

42. The method of claim 37, further including specifying, with the 128 different moving object definitions, moving object characters of 12 different sizes.

43. The method of claim 37, further including writing rotation/scaling parameters to the object attribute memory storage.

44. The method of claim 37, further including writing mosaic information to the object attribute memory storage.

45. The method of claim 37, further including displaying plural display windows simultaneously.

Sub C1 A1

46. The method of claim 37, further including independently controlling the scrolling of plural display windows.

47. The method of claim 37, further including independently controlling the rotation of plural display windows.

48. The method of claim 37, further including independently controlling the alpha blending of plural display windows.

49. The method of claim 37, further including independently controlling the fade-in/fade-out of plural display windows.

Sub C2

50. The method of claim 37, further including independently controlling the upper-left and lower-right display portions of plural display windows.

51. The method of claim 37, further including performing arithmetic operations on two selected surfaces and processing for up to 16 levels of semi-transparency.

52. A method of monitoring the status of video game play comprising:

reading a 16-bit value from a predetermined memory location;

decoding a bit position 00 of the 16-bit value to determine vertical blanking interval status;

decoding a bit position 01 of the 16-bit value to determine horizontal blanking interval status;

decoding a bit position 02 of the 16-bit value to determine vertical counter matching or non-matching;

decoding a bit position 03 of the 16-bit value to determine whether vertical blanking interval interrupts are enabled;

decoding a bit position 04 of the 16-bit value to determine whether horizontal blanking interval interrupts are enabled; and

decoding a bit position 05 of the 16-bit value to determine whether vertical counter matching interrupts are enabled.

53. The method of claim 52, wherein the predetermined memory location is 004h.

54. A method of controlling interrupts in a video game playing system comprising:

writing a value to bit position 03 of a 16-bit word at a predetermined memory location to

specify whether vertical blanking interval interrupts are enabled;

writing a value to bit position 04 of the 16-bit word to specify whether horizontal blanking interval interrupts are enabled; and

writing a value to bit position 05 of the 16-bit word to specify whether vertical counter matching interrupts are enabled.

55. The method of claim 54, wherein the predetermined memory location is 004h.

56. A method of controlling display status in a video game playing system comprising:

writing a three-bit background mode specifier to the bit positions 00-02 of a 16-bit word at a predetermined memory location;

writing a display frame selector bit to bit position 04 of the 16-bit word at the predetermined memory location to select between two different frame buffers;

writing a control bit to bit position 05 of the 16-bit word at the predetermined memory location to select whether to render objects during horizontal blanking intervals;

writing a control bit to bit position 06 of the 16-bit word at the predetermined memory location to select between one-dimensional and two-dimensional object character mapping;

writing values to bit positions 08-12 of the 16-bit word at the predetermined memory location to select display of four different background screens and display of moving objects;

writing values to bit positions 13-14 of the 16-bit word at the predetermined memory location to select display of two different windows; and

writing values to bit position 15 of the 16-bit word at the predetermined memory location to select display of an object window.

57. The method of claim 56, wherein the predetermined memory location is 0000h.

58. A method of generating a video game display based on a display instruction comprising:

decoding a three-bit background mode specifier at bit positions 00-02 of a 16-bit display instruction written to a predetermined memory location to determine background mode;

decoding a display frame selector bit at bit position 04 of the 16-bit display instruction written to the predetermined memory location to select between two different frame buffers;

A1

decoding a control bit at bit position 05 of the 16-bit display instruction written to the predetermined memory location to select whether to render objects during horizontal blanking intervals;

decoding a control bit at bit position 06 of the 16-bit display instruction written to the predetermined memory location to select between one-dimensional and two-dimensional object character mapping;

decoding bit positions 08-12 of the 16-bit display instruction written to the predetermined memory location to select display of four different background screens and display of moving objects;

decoding bit positions 13-14 of the 16-bit display instruction written to the predetermined memory location to select display of plural windows;

decoding bit position 15 of the 16-bit display instruction written to the predetermined memory location to select display of a moving object window; and

generating a display based on said decoding steps.

59. The method of claim 58, wherein the predetermined memory location is 0000h.

60. A method of generating a video game display based on a display instruction comprising:

decoding a two-bit priority specification at bit positions 00-01 of a 16-bit value written to location 008h or 00Ah specifying one of four priority levels;

decoding a two-bit character base block at bit positions 02-03 of the 16-bit value written to location 008h or 00Ah specifying a character base block value;

A1

decoding a mosaic enable/disable flag at bit position 06 of the 16-bit value written to location 008h or 00Ah;

decoding a color mode selector at bit position 07 of the 16-bit value written to location 008h or 00Ah to select between a 16 color, 16 palette color mode and a 256 color, one palette color mode;

decoding a screen base block at bit positions 08-12 of the 16-bit value written to location 008h or 00Ah;

decoding a screen size specifier at bit positions 14-15 of the 16-bit value written to location 008h or 00Ah; and

generating a display based on the decoding steps.

61. In a system for allowing a person to play a video game by interactively displaying graphical information on a display in response to interaction of the person with at least one human-manipulable control, a mosaic control method comprising:

(a) selecting a mosaic object display mode; and

(b) writing a 16-bit mosaic control value to an address 04Ch, the mosaic control value including a background character mosaic horizontal size in bit positions 00-03, a background character mosaic vertical size in bit positions 04-07, a moving object character mosaic horizontal size in bit positions 08-011 and a moving object character mosaic vertical size in bit positions 12-15,

wherein the mosaic sizes specify how many dots in an original character should be replaced by a virtual character.

A1

62. In a system capable of video game play that displays graphical information on a display at least in part in response to user manipulation of a control, a method of controlling rotation and/or scaling comprising:

decoding a 12-bit value specifying an x-coordinate reference starting point;

decoding a 12-bit value specifying a y-coordinate reference starting point;

decoding a 16-bit value specifying the distance of movement in the x direction;

decoding a 16-bit value specifying the distance of movement in the y direction;

performing rotation and/or scaling processing of stored background data using said above-mentioned values; and

displaying a resulting rotated and/or scaled image on the display.

63. In a system capable of playing a video game by displaying graphical information at least in part in response to real time user manipulation of at least one user control, a method of decoding a windowing command comprising:

decoding bit positions 00 through 04 of a 16-bit value at location 048h to determine whether or not to display any or all of four background displays and a moving object display in a first display window;

decoding a bit at bit position 05 of the 16-bit value to select whether or not to enable color special effects within the first display window;

decoding bits at bit positions 08-12 of the 16-bit value to determine whether or not to display any or all of the four background displays and the moving object display within a second display window different than the first display window; and

decoding a bit at bit position 13 of the 16-bit value to determine whether to enable color special effects within the second window.--